NUMERICAL GRID GENERATION
Joe F. Thompson, editor

ADAPTIVE GRIDDING FOR FINITE DIFFERENCE SOLUTIONS TO HEAT AND MASS TRANSFER PROBLEMS*

HARRY A. DWYER†, MITCHELL D. SMOOKE‡ AND ROBERT J. KEE‡
†University of California, Mechanical Engineering Department, Davis, CA 95616, USA; ‡Sandia National Laboratories, Applied Mathematics Division, Livermore, CA 94550, USA

INTRODUCTION

Our purpose in writing this paper is to review some of our recent work in the calculation of optimal meshes for the solution of parabolic and elliptic partial differential equations (PDE). We first explain our strategies for the adaptive placement of mesh points. In addition, we make some speculation as to promising avenues for future research in mesh adaptation. Finally, we discuss examples of the application of adaptive gridding to problems of heat and mass transfer. We draw these examples from our work in combustion modeling.

In obtaining numerical solutions to PDEs, the spatial derivatives are often approximated by discrete representations on a mesh network. The accuracy of any numerical solution depends in an important way on the relationship of the location of the mesh points to changes in the dependent variables. Our objective is to investigate finite difference methods in which the mesh networks adapt themselves dynamically to obtain accurate solutions. Such methods represent an important advance in overcoming a major shortcoming of traditional fixed mesh methods which are often unable to resolve accurately steep fronts or sharp peaks.

Our research in adaptive meshing follows two avenues. One is to employ a fixed number of mesh points and to move their location by coordinate transformations. The other is to add or subtract mesh points as needed. In either case the positioning of the mesh depends on one or more important characteristic of the solution. We attempt to equidistribute this characteristic between each mesh interval. For example, equidistribution of the arc-length of the solution has the effect of concentrating mesh points in steep gradients. Taking the coordinate transformation approach, the original equations are recast so that the new independent variable becomes the arc-length. Then, in addition to solving the original equations in the transformed variables, a set of equations is also solved to describe the movement of the original physical coordinates relative to the new transformed variables. When adding and subtracting grid points (the variable node approach) we specify the maximum value of the equidistributed characteristic (say arc-length) allowed over any mesh interval, and continue to add points until this criterion is satisfied. The latter approach is closer to that used in ODE software where as many timesteps as needed are

* Research sponsored by the United States Department of Energy, Office of Basic Energy Sciences.

taken to bring the local truncation error to within prespecified tolerances. However, while this approach may be needed to insure high accuracy for PDEs, it can suffer from limitations of computer storage and time. While we are developing two approaches for adaptive meshing, we believe that the research will ultimately lead to a combination of the methods.

Our variable node method stems largely from our development of methods to solve systems of stiff and unstable nonlinear boundary value problems. Such systems occur frequently in modeling energy systems. Our applications have been principally in combustion chemistry, particularly in the investigation of complex chemical kinetics in premixed flames. Our coordinate transformation work was initially used to track moving flame fronts, and more recently to investigate droplet combustion. In all our work we are concerned with solving simultaneously a relatively large number of PDEs; in the case of the premixed flames, 30 coupled PDEs are typical.

Our work draws on earlier work in both the aeronautical and the boundary value problem literature. From the former we take the ideas of generalized non-orthogonal coordinate transformations and boundary-fitted coordinate systems.[1,2] From the latter we take the ideas of equidistribution of weight functions and error control strategies. Generally speaking, the boundary value problem literature has more theory on which to base methods, but the problems are simpler inasmuch as they are one-dimensional.

BASIC SYSTEM OF EQUATIONS

The solutions to the physical problems which are presented in this paper cover a range of flow and chemical systems. However, in all of the problems there is the common simplification of uncoupling the fluid mechanics from the heat and mass transfer. For some systems, such as steady flame propagation, the simplification is natural to the problem, while in others, it is more artificial. In either case, it does allow for a clear understanding of the problems caused in grid adaptation, when heat and mass transfer as well as chemical reactions are considered.

The system of reaction-diffusion equations which describe the problems in this paper are:

$$\frac{\partial}{\partial t}(\rho Z_m)+\frac{\partial}{\partial x}(\rho u Z_m)+\frac{\partial}{\partial y}(\rho v Z_m)=\frac{\partial}{\partial x}(D_m\frac{\partial Z_m}{\partial x})+\frac{\partial}{\partial y}(D_m\frac{\partial Z_m}{\partial y})+\dot{\omega}_m \qquad (1)$$

where Z is the dependent variable vector (temperature, and species mass fractions), and $\dot{\omega}_m$ is the vector representing the the chemical source terms:

$$Z=(T,Y_1,Y_2,\ldots,Y_K)^t \qquad (2)$$

$$\dot{\omega}=(\dot{\omega}_T,\dot{\omega}_1,\dot{\omega}_2,\ldots\dot{\omega}_K)^t \qquad (3)$$

In these equations the following notation has been employed: ρ - mass density, u - velocity in x-direction, v - velocity in the y-direction and D_m - the diffusive transport coefficient for

the m^{th} equation. The details of the source terms and velocity fields are described when the physical examples and results are presented later in the paper.

For some of the results presented, the above equation set is transformed into generalized non-orthogonal coordinates (τ, ξ and η):

$$\frac{\partial \hat{Q}}{\partial \tau} + \frac{\partial \hat{E}}{\partial \xi} + \frac{\partial \hat{F}}{\partial \eta} = \frac{\partial \hat{R}}{\partial \xi} + \frac{\partial \hat{S}}{\partial \eta} + \hat{H} \tag{4}$$

where,

$$\hat{Q} = \frac{Z_m}{J}$$

$$\hat{E} = \frac{Z_m}{J}(\xi_t + \rho u \xi_x + \rho v \xi_y)$$

$$\hat{F} = \frac{Z_m}{J}(\eta_t + \rho u \eta_x + \rho v \eta_y)$$

$$\hat{R} = \frac{D_m}{J}\left(\frac{\partial Z_m}{\partial x}\xi_x + \frac{\partial Z_m}{\partial y}\xi_y\right)$$

$$\hat{S} = \frac{D_m}{J}\left(\frac{\partial Z_m}{\partial x}\eta_x + \frac{\partial Z_m}{\partial y}\eta_y\right)$$

$$\hat{H} = \frac{\dot{\omega}_m}{J}$$

The transformation metrics, or areas and volumes, are given by:

$$J = \frac{1}{x_\xi y_\eta - x_\eta y_\xi}$$

$$\xi_x = J y_\eta, \qquad \xi_y = -J x_\eta, \qquad \xi_t = -x_t \xi_x - y_t \xi_y$$
$$\eta_x = -J y_\xi, \qquad \eta_y = J x_\xi, \qquad \eta_t = -x_t \eta_x - y_t \eta_y$$

We readily see that the resulting equations are more complex; however, with some additional programming a much more valuable tool is obtained. In the above form it is quite easy to implement body-oriented coordinates for arbitrary-shaped bodies, as is often done for flow over airfoils. However, the major advantage of these transformations in our work is the ease with which coordinate adaptation can be utilized.

Even though the governing equations are somewhat simplified compared with the Navier-Stokes equations, they still encompass a large selection of important problems. Moreover, they include a rich and disparate collection of physical time scales. As a result, they are likely to have solutions with regions which need adaptive gridding to be resolved accurately. For example, the effects of the following time scales will be illustrated in the results presented

$\Delta t_U = L/U$, Velocity convection scale

$\Delta t_\nu = L^2/\nu$, Viscous diffusion scale

$\Delta t_\alpha = L^2/\alpha$, Heat conduction scale

$\Delta t_{D_m} = L^2/D_m$, Mass diffusion scale

$\Delta t_{\dot{\omega}_m}$ = Reaction rate scale

As these scales become disparate (depending on the particular problem), steep gradients in space and time develop. Without adaptation, the numerical integration methods can be pushed to dramatic failure. It is our purpose to present methods which resolve these scales in space and time in an efficient and accurate manner.

ADAPTIVE GRID METHODS

We consider both steady and transient problems. The steady problems are elliptic boundary value problems, while the transient problems are parabolic initial-boundary value problems. At each time step of a transient problem, however, an elliptic boundary value problem must be solved. Therefore, our meshing strategies share the same essential features regardless if the application is steady or time-dependent.

During the last fifteen years many varied methods have been developed to attempt to choose optimal grid spacings on which to solve two-point boundary value problems. When these problems are solved using an initial value method (such as shooting), the adaptive meshing is done automatically and accurately. Variable-step initial value problem software is used to adjust the step size as the integration proceeds in order to control the local truncation error. Unfortunately, many problems in combustion are unstable to initial value methods,[3] and therefore global solution methods must be used.

Kautsky and Nichols[4] point out that many of the adaptive mesh selection procedures used for global solution methods can be interpreted as equidistributing a positive weight function. On the interval [0,L], one attempts to determine a mesh M

$$M = (0 = x_1 < x_2 < \ldots < x_M = L)$$

such that the weight function achieves a given constant value over each subinterval. Among the various approaches developed, White[5] has discussed equidistribution of the arc-length of the solution; Pereyra and Sewell[6] have equidistributed the local truncation error and Smooke[3] has chosen to equidistribute both the change in the discrete solution and its gradient. Other methods for choosing appropriate meshes for two-point boundary value problems have been investigated, for example, by Russell and Christiansen,[7] Ablow and Schecter,[8] de Rivas,[9] and Denny and Landis.[10]

Positive Weight Function Concept

Following the notation of Kautsky and Nichols, we say that the mesh M is equidistributed on [0,L] with respect to the non-negative weight function f and the constant W if

$$\int_{x_j}^{x_{j+1}} f \, dx = W, \quad j = 1,2,\ldots,M-1 \tag{5}$$

Similarly M is called sub-equidistributing on [0,L] with respect to f and W if

$$\int_{x_j}^{x_{j+1}} f\, dx \leq W, \quad j = 1, 2, \ldots, M-1 \tag{6}$$

Strategies for determining an optimal mesh for two-point boundary value problems can be implemented either implicitly or explicitly. In the implicit approach, the weight function depends directly upon the solution. As a result, the original boundary value problem is converted into an augmented system in which the dependent variables and the mesh are computed simultaneously. In the explicit approach, the weight function does not depend upon the current solution. Instead, it depends upon a previously calculated solution. For both linear and nonlinear boundary value problems, the implicit approach requires that one solve a nonlinear two-point boundary value problem. Thus implicit equidistribution techniques do not preserve the linear-nonlinear character of the original problem. Moreover, even for nonlinear problems the augmented system is usually more difficult to solve than the original problem. Explicit equidistribution techniques, on the other hand, preserve the linear-nonlinear character of the original two-point boundary value problem.

Our experience has led us to consider explicit equidistribution methods. We have found that as the number of dependent variables increases, or the problem becomes more nonlinear, the selection of a mesh by equidistributing an implicit weight function is less practical than by equidistributing a weight function based on the solution from a previous grid. The approach we have chosen to determine an adaptive grid for premixed flame problems is similar to the method used by Pearson[11] in solving scalar boundary layer problems. We attempt to equidistribute the difference in the components of the discrete solution and the difference in the gradient of the components of the discrete solution between adjacent mesh points. That is we seek to obtain a mesh M such that

$$\int_{x_j}^{x_{j+1}} \left|\frac{dZ_i}{dx}\right| dx \leq \delta\left(\max_{0\leq x\leq L} |Z_i|\right) \quad \begin{matrix} j = 1, 2, \ldots, M-1 \\ i = 1, 2, \ldots, K+1 \end{matrix}, \tag{7}$$

and

$$\int_{x_j}^{x_{j+1}} \left|\frac{d^2Z_i}{dx^2}\right| dx \leq \gamma\left(\max_{0\leq x\leq L} \left|\frac{dZ_i}{dx}\right|\right) \quad \begin{matrix} j = 1, 2, \ldots, M-1 \\ i = 1, 2, \ldots, K+1 \end{matrix}, \tag{8}$$

where Z is the dependent variable vector, δ and γ are small numbers less than one and the values of $\max|Z_i|$ and $\max|dZ_i/dx|$ are obtained from a converged numerical solution on a previously determined mesh.

A potential disadvantage of the method described so far is that the mesh may not be smoothly varying. For example, the ratio of consecutive mesh intervals may differ by several orders of magnitude. This can adversly affect the accuracy of the solution as well as the convergence properties of the method. As a result, we impose the added constraint that the mesh be locally bounded, i.e., the ratio of adjacent mesh intervals must be bounded above and

below by constants. We require that

$$\frac{1}{C} \leq \frac{h_j}{h_{j-1}} \leq C, \quad j = 2,3,\ldots,M \tag{9}$$

where $h_j = x_j - x_{j-1}$ and C is a constant less than one. Such a "buffering" of the mesh tends to smooth out rapid changes in the size of the mesh intervals.

We note here an analogy to the approach followed for time step selection in predictor-corrector ODE software. In these codes some estimate of the local truncation error is made. One way to measure the error is by comparing the difference between a certain order predictor and the same order corrector. Their difference is related to the local truncation error incurred in taking a time step. The time step is then adjusted such that this error is below a prespecified level. Possible ways to measure truncation error are to use different differencing formulas, or to use the same difference formulas but on different meshes.

Certainly the equidistribution and control of local truncation error is the most conservative and accurate approach to mesh adaptation. However, in many cases it may be more costly than necessary. For instance, if only integrated properties of the solution are of interest (e.g. flame speed or surface drag) then perhaps less attention need be paid to truncation error everywhere in the flow field. For problems with strong nonlinearities it may even be preferable to equidistribute something related to the local truncation error rather than the error itself. In particular, we have seen that weight functions based on higher derivatives (to more closely match truncation error) have led to instabilities. Moreover, if the differencing scheme is first order then the local truncation error is proportional to second derivatives of the solution. Thus, the weight function in Eq. (8) is proportional to the local truncation error. As a result we believe that weight functions similar to those in Eqs. (7), (8) and (10), are perfectly adaquate for many problems.

Steady-State Problems, Variable Node Method

After discretization, the governing equations form a nonlinear system of algebraic equations. We solve this system by a damped modified Newton method.[12] First the equations are solved on a uniformly spaced coarse mesh (3-5 subintervals). The values of $\max|Z_i|$ and $\max|dZ_i/dx|$ are then evaluated. We next test the inequalities in Eq. (7) and Eq. (8) for each of the $K+1$ components of Z at each node of the coarse mesh. If either of the inequalities is not satisfied, a grid point is inserted at the midpoint of the interval in question. Once a new mesh has been obtained, we check to see whether it is locally bounded. If it is not, a grid point is inserted at the midpoint of the intervals in which the inequality in Eq. (9) is not satisfied. The previously converged numerical solution is interpolated onto this new mesh and the result serves as an initial solution estimate for Newton's method on this finer grid. The governing equations are solved on the new mesh and the process continues until the inequalities in Eqs.

(7), (8) and (9) are satisfied. Note that if we had refined the mesh by using only the inequality in Eq. (7), we would resolve high gradient regions but would have difficulty resolving regions of high curvature (for example the local maxima of sharp peaks in the solution).

Most of the ideas discussed above can be extended to the solution of multi-dimensional elliptic boundary value problems. We are just beginning to explore methods to obtain optimal grids for two-dimensional nonlinear elliptic boundary value problems. In our initial attempts we have made the logical extension of the one-dimensional ideas. That is, we start on a coarse two-dimensional grid, and add grid pionts according to Eqs. (7), (8) and (9) in both the x and y directions. In two-dimensions the Jacobian of the system is block penta-diagonal and we solve the system by block line SOR iteration. If fronts in the solution align reasonably well with one of the coordinates then the method is efficient. Of course if a front crosses the mesh on a bias then a fine mesh results everywhere and the direct extension of the one-dimensional method is not really useful. In these cases either a coordinate rotation or a local mesh refinement[13] must be employed.

Time-Dependent Problems

The ideas used in solving one-dimensional steady-state problems are readily adapted for time-dependent mixed initial-boundary value problems. In particular, by considering the solution of a time dependent problem as the solution of an inhomogeneous two-point boundary value problem at each time level, the methods developed in the realm of steady-state problems can be applied in a time-dependent setting. Both the implicit and the explicit equidistribution techniques have natural time-dependent analogues. In the case of the implicit methods, the original equations are recast so that in addition to solving the original equations in the transformed variables a set of equations is also solved to describe the movement of the original physical coordinates relative to the new transformed variables. In general, one can expect the difficulty with specifying an initial solution estimate for the dependent solution components and the grid points to be less severe in the time-dependent setting than in the steady-state one since the previous time level solution is often an excellent starting estimate. The explicit equidistribution techniques can be used in a time-dependent setting by explicitly moving the grid based upon solutions at previous time levels.

Coordinate Transformation Methods

Our coordinate transformation technique has been tested extensively by Dwyer, et. al. for one-dimensional problems, and more recently, it has also performed quite well in two dimensions.[14] We note, however, that so far we have not implemented a general two-dimensional adaptation procedure. Instead, we take advantage of some *a priori* physical knowledge of the

solution to extend the one-dimensional method. Although the method is not yet fully adaptive in two-dimensions, we believe that the solution to many important problems can benefit from its use. Moreover, we believe that generalization of the method will follow from current work. The solution technique in the two-dimensional problems is a non-iterative block-tridiagonal ADI method[15,16] in which the Jacobians are evaluated analytically.

In this method the lines of constant η are fixed (forming arcs in space), and the adaptation is done along these arcs. In effect, the method is quasi-one-dimensional, and it relies on the modeler having sufficient qualitative knowledge about the solution to be able to fix a set of coordinates which are roughly normal to any steep fronts in the solution. We typically take the weight function (or transformation) for adaptation along the fixed arcs to be given by:

$$\xi(x,y,t) = \frac{\int_0^S (1 + b|\partial T/\partial S|)dS}{\int_0^{S_{max}} (1 + b|\partial T/\partial S|)dS} \tag{10}$$

where S is the length along the fixed arc, and b is an adjustable constant used for "optimization" of the grid distribution. For the case $b = 0$ a uniform distribution of points along the fixed arc results. For large values of b, the mesh intervals are determined so that the same change in the dependent variable T occurs between mesh points. A typical value of b is 1/3. The coefficient b can be thought of in terms of the "buffering" concept introduced earlier. That is, b is chosen so that not all the mesh points are concentrated in the front region. Some are in regions of relatively uniform T, and there is a smooth progression of mesh interval sizes in moving away from a front.

The weight function is evaluated explicitly, and the mesh transformation is held fixed throughout the time step. In some cases, however, we have used a prediction of the solution at $t + \Delta t$ instead of the solution at t to form the basis of the transformation. In all cases the integrals in Eq. (10) are evaluated using the trapezoidal rule. If Δt is large enough for the front to move out of the fine-mesh region, then implicit or iterative coordinate generation schemes would have to be considered. However, this was never the case in our problems, since the fast chemical reactions prohibit the taking of large time steps. Also, the buffering effect of the b parameter causes there to be adaquate resolution even if the front does move away from its optimal location.

Linear Algebra Considerations

We take as an assumption that for problems of interest (in combustion) the system of equations is stiff—they are characterized by widely disparate time and length scales. This fact leads us to consider only implicit solution procedures.[17] A salient characteristic of implicit methods is that they require the simultaneous solution of nonlinear equations at each time step (or iteration). For multi-dimensional problems or problems involving many dependent

variables (e.g. species concentrations), these solutions lead to a need to form and solve systems represented by large matrices. Therefore, the way in which this task is accomplished has a major bearing on the structure of the computer codes which solve the systems.

Our approach currently for one-dimensional problems is to employ a modified Newton method. The block tri-diagonal Jacobian matrix is formed numerically using finite differences and its LU decomposition is computed immediately. The LU decomposition factors occupy the same storage locations as the Jacobian did originally. The same decomposed matrix is then used for several iterations (or time steps in transient problems). As long as the iterations are convergent, a sizable cost savings is realized by not re-evaluating and factoring the Jacobian. This approach is commonly used for solving systems of stiff nonlinear ODEs.

In two-dimensional problems we take two approaches. For the fixed-number-of-grids coordinate transformation problems we employ a standard alternating direction implicit (ADI) method. Here the block tri-diagonal Jacobian is formed and LU decomposed, and the linear system is solved along each row and column of the mesh at each time step. No iteration is done. Justification for the approach follows the well-known arguments that the error incurred by the ADI splitting is of the same order as the truncation error already incurred by the discretization of the time derivative.[15,16]

We take a different approach in solving the nonlinear equations in the variable node formulation. Here the full Jacobian, a block five-diagonal matrix, is formed at once. A modified Newton method is used to solve the nonlinear system. At each stage of the Newton iteration an iterative block-line-SOR method is employed to solve the linear system. The LU factors are stored and re-used for successive iterations. However, after the solution is completed on a given mesh and new mesh points are added as needed, a new Jacobian must be computed on the new mesh.

We expect that significant computational gains will result from research on and development of incomplete Jacobian factorizations or matrix splittings. The objective here is to avoid solving the original equations directly, and instead to solve a related, and approximately equivalent, system that is much easier to solve. The best known example of such splitting is the ADI method, which can be thought of as an incomplete factorization of the full Jacobian. Even though the factorization is incomplete, the error which it introduces is of the same order as that introduced by the time discretization. Therefore, the approximation does not degrade the accuracy of the solution but it increases significantly the efficiency of the computation.

The ADI factorization is only one of a large family of related splittings which can take advantage of some particular characteristic of a problem. For example, it is often the case in systems of PDEs that some of the equations are weakly coupled to the others. In such cases, solving the equations sequentially (instead of fully coupled) is known to result in significant savings. Instead of solving systems of block tri-diagonal equations, one is able to solve a sequence of scalar tri-diagonal equations with far fewer operations required. Similarly, in some

combustion problems, considerable savings are realized through operator splitting algorithms in which the chemical rate terms are handled separately from the transport terms. These methods are equivalent to matrix splittings of the system's Jacobian. However, in both cases, application of the procedures has been ad-hoc, i.e. with little theory to help determine the rate of convergence, or whether the process converges or not. By studying pre-conditionings and incomplete factorizations of the Jacobians, rather than ad-hoc splittings of the equations, such methods can be put on a firmer theoretical footing and thus more reliable and effective PDE methods should result.

The cost of evaluating the Jacobian is usually very high in our problems (up to 95% of the computer time in some flame problems). Therefore, it is natural to seek methods which require as few Jacobian evaluations as possible. Based on the success of the modified Newton method,[18] where we have applied it, and its success in the ODE software, we expect that similar approaches will ultimately find wider application in the solution of PDEs. The dilemma is that in order to use a modified Newton method, the full Jacobian must be stored. For multi-dimensional problems this storage requirement is usually too large for the memory of any computer in use today. Therefore, effective use of a modified Newton method requires development of algorithms which quickly move Jacobian information between computer memory and peripheral storage. We note here also that some splitting methods, as discussed above, lead to fewer function evaluations to complete a Jacobian evaluation.

EXAMPLE PROBLEMS

Steady Premixed Flames

The method we have implemented in the calculation of premixed flame structure equi-distributes the difference in the components of the solution and its gradient between consecutive grid points. To illustrate the importance of adaptively placing grid points in the flame zone to the accuracy and efficiency of the flame calculation, we have performed several calculations for an acetylene-oxygen flame using equi-spaced and adaptively placed grids. (For these problems a system of 21 species and 72 reactions was used.) Figure 1 shows the molecular hydrogen profiles for a series of calculations using 20, 40, 80, and 160 equi-spaced points. We include the experimental data for reference. We secure not only a much smoother solution but one which agrees better with the experimental data as a finer and finer grid is used.

Figure 2 shows the molecular hydrogen profile for the same flame but solved using adaptive meshing. In this case 41 adaptively placed points are used to obtain three significant figures of accuracy in the solution. As expected, the adaptive calculation secures a highly resolved species profile with far fewer points than are required using the equi-spaced grid.

In the adaptive calclation, 19 of the 41 grid points are located in the "flame zone," or region of fast chemical reaction. Note that a relatively large region of the computation has

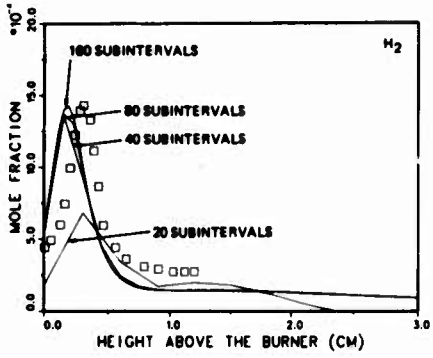

Figure 1. Hydrogen mole fraction distribution profiles in an acetylene-oxygen flame, computed on various equi-spaced grids. Boxes are experimental data of Eberius, Hoyermann and Wagner.

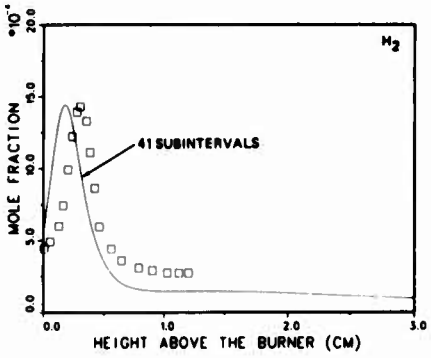

Figure 2. Hydrogen mole fraction distribution profiles in an acetylene-oxygen flame, computed on 41 adaptively placed grids. Boxes are experimental data of Eberius, Hoyermann and Wagner.

relatively little chemical reaction. Either the temperature is too low, or the fuel is almost all consumed. The smallest mesh interval is 640 times smaller than the total interval of the problem. So to obtain the same resolution over 600 equi-spaced grids would be required. The adaptive calculation took 275 seconds of CPU time on a CRAY-1S computer. The equi-spaced calculation with 160 subintervals took 585 seconds of CPU time.

In the next example we compare the effects of adaptive and equi-spaced grids in the prediction of flame speeds in a one-atmosphere, stoichiometric, hydrogen-air flame.[19] The accurate placement of grid points in regions where the solution varies rapidly leads to a significant reduction in the number of subintervals needed to obtain accurate flame speeds. As a result, the overall cost of a flame speed computation can be substantially reduced. In the first set of calculatons we determined flame speeds on grids consistng of 20, 40, 80, 160, 320, and 640 equi-spaced points. The results of the calculatons are listed in Table I.

The second set of calculations was performed using the adaptive grid procedure. In this case we used grids of 20, 30, 40, 50, and 60 adaptively placed points. The results are listed in Table II.

Several points merit further discussion. First, for both the equi-spaced and adaptively placed grids, we see that as the number of mesh intervals increases, the flame speeds decrease. Second, the sequence of flame velocities obtained in the adaptive calculations approach a limiting value with only 40 to 50 grid points, while flame velocities obtained in the equi-spaced calculations are still changing by almost 15 percent as we go from 80 to 160 grid points. In fact, it was not until 640 equi-spaced points were used that the flame speed was within 2 percent of the result calculated on the 50 point adaptive grid. Like the previous example, the ratio of

TABLE I

HYDROGEN-AIR FLAME SPEEDS, EQUI-SPACED GRIDS (cm/sec)

No. of Points	20	40	80	160	320	640
Flame Speed	445	289	244	211	193	184

TABLE II.

HYDROGEN-AIR FLAME SPEEDS, ADAPTIVE GRIDS (cm/sec)

No. of Points	20	30	40	50	60
Flame Speed	248	212	185	181	181

minimum mesh size to the domain of integration was 625. Also, as expected, the adaptive grid computation is less expensive. The 50 point adaptive calclation took 45 seconds of CPU time while the 640 point equi-spaced calculation took 327 seconds. A savings of about a factor of seven resulted in going from equi-spaced to adaptive grids.

Two-Dimensional Elliptic Boundary Value Problem

We demonstrate here our two-dimensional extension of the variable node method. The equation we have chosen is the nonlinear Poisson equation on the unit square:

$$\frac{\partial^2 Z}{\partial x^2} + \frac{\partial^2 Z}{\partial y^2} + Z^2 = f(x,y)$$

$$Z = g(x,y) \text{ on the boundary}$$

We have chosen $f(x,y)$ and $g(x,y)$ so that the solution is $Z = \exp -30(x^2 + y^2)$. The initial equi-spaced grid was 2×2. After five mesh refinements the nonuniform 18×18 mesh shown in Fig. 3 evolved. Note the high resolution of the solution in the regions of high slope and curvature.

Unsteady Two-Dimensional Flame Propagation, Coordinate Transformation Method

In this section we demonstrate coordinate transformation adaptive grid techniques by disccussing several examples. First we present solutions for unsteady flame propagation about spherical particles. In these examples the time scales for convection and reaction are small

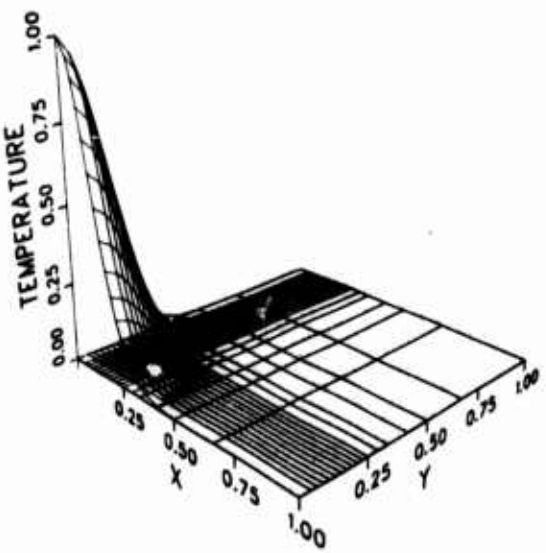

Figure 3. Solution to the elliptic test problem with a two-dimensional variable node adaptive grid.

compared to conduction and diffusion, or

$$\Delta t_U \text{ and } \Delta t_{\dot{\omega}_m} \ll \Delta t_\alpha \text{ and } \Delta t_{D_m}$$

A one-step chemical reaction is used and the vector of dependent variables and rate terms are

$$Z_m = (T, \rho_A)$$

$$\dot{\omega}_m = \left(\rho_A \frac{\Delta t_\alpha}{\Delta t_{\dot{\omega}_A}} \exp(-\theta_A/T), -\rho_A \frac{\Delta t_\alpha}{\Delta t_{\dot{\omega}_A}} \exp(-\theta_A/T)\right)$$

where T, ρ_A and θ_A, the nondimensional temperature, premixed fuel concentration and activation energy, have been normalized by reference values.[14] The calculation is simplified so that the overall density remains constant and thus the flow field is independent of the combustion process. The velocity field is given as a low Reynolds number Stokes flow.

The results of an interesting calculation are shown in Figs. 4 through 9. The following ratios of time scales are used:

$$\frac{\Delta t_\alpha}{\Delta t_\nu} = \frac{\Delta t_{DA}}{\Delta t_\nu} = Pe \text{ (Peclet Number)} = 200$$

$$\frac{\Delta t_\alpha}{\Delta t_{\dot{\omega}_A}} = 2.2 \times 10^5$$

Figures 4 and 5 illustrate unsteady flame propagation after surface ignition, when using a uniform grid. (These figures are divided into two parts, the top shows the coordinate system

and the bottom plots the isotherms. There are ten normalized isotherms plotted which range in values between 0.2 and 1.2) The figures show that the grid is uniform and the isotherm distribution exhibits significant oscillation. Note that the oscillation in the isotherms becomes larger in amplitude as the flame moves into the large cell regions away from the body. These oscillations are a result of the large cell Peclet number and the central difference approximation for the spatial derivatives.[20] The cell Peclet number is large because of the increasing velocity and cell size as the grid moves away from the body. If we had used windward differences, the numerical viscosity would have increased significantly, and thus introduce significant errors such as an the increase in flame thickness. Use of a refined uniform grid is unreasonable because of the additional computational requirements of time and storage.

Now consider the problem using an adaptive grid as shown in Figs. 6 and 7. These figures show the coordinate and isotherm distributions for the same times as shown in Figs. 4 and 5. Notice that the flame has a new and more accurate velocity and position and that there are no oscillations. By resolving the flame, the cell Peclet number is reduced to values less than one. This guarantees that the solution will be oscillation free. To illustrate our point further, the radial temperature distributions at similar angular positions are shown in Fig. 8 for both the uniform and adaptive grid solution. The oscillations in the uniform grid solution are quite

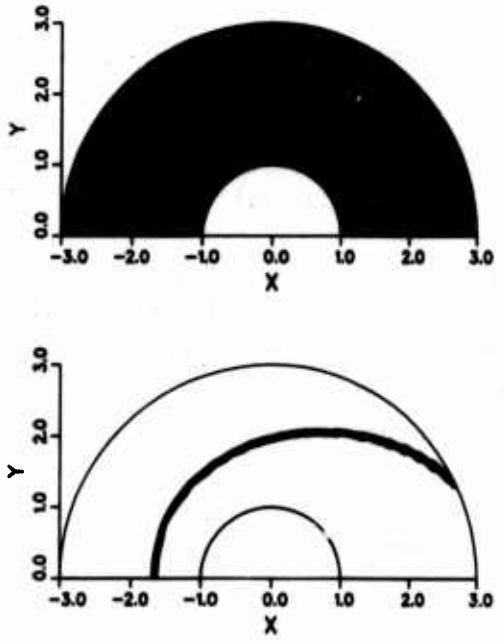

Figure 4. Coordinate system and isotherm distribution about a burning particle with a uniform grid, early time.

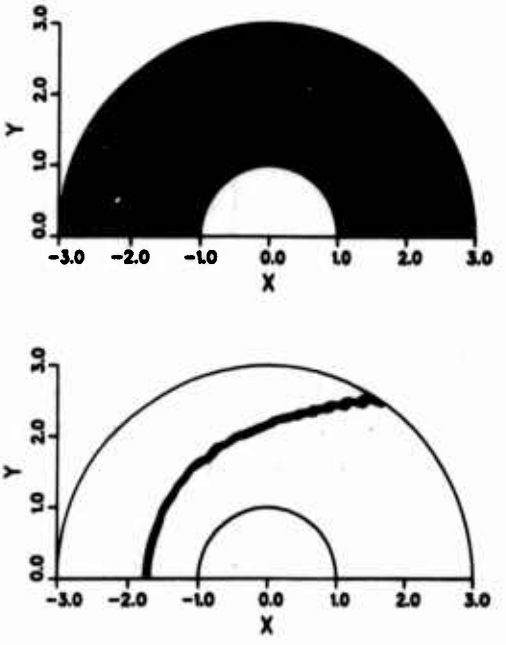

Figure 5. Coordinate system and isotherm distribution about a burning particle with a uniform grid, later time.

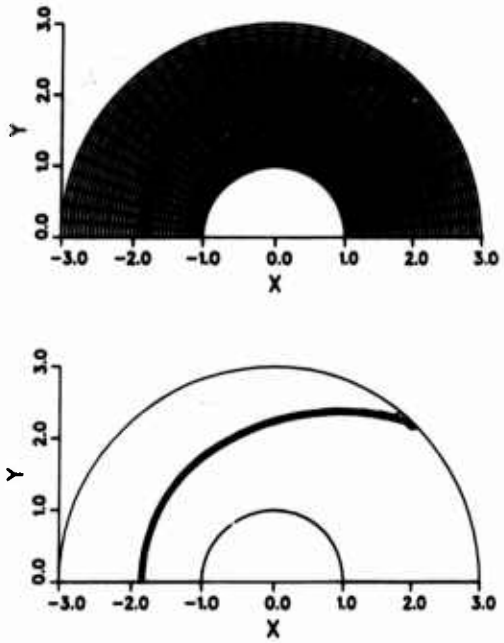

Figure 6. Coordinate system and isotherm distribution about a burning particle with a coordinate transformation adaptive grid, early time.

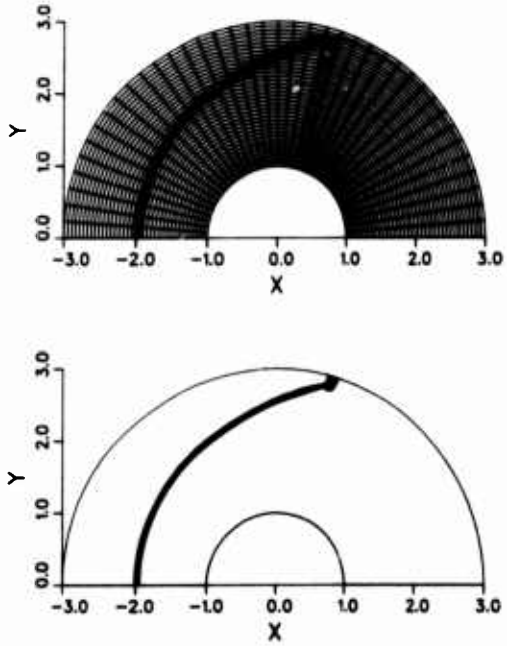

Figure 7. Coordinate system and isotherm distribution about a burning particle with a coordinate transformation adaptive grid, later time.

apparent. Also, it should be mentioned that the uniform grid solution terminated at the next time step because of negative temperatures caused by the oscillations.

With the same number of grid points we have been able to convert an unusable calculation to an efficient and accurate one. However, we have introduced some new, but minor, problems with the remedy. One of these problems is caused when the thin flame passes out of the boundaries of the system and there are no longer any gradients along some of the fixed arcs. The grid then reverts back to a uniform grid over one time step. In the present calculation this does not cause a problem because the dependent variable is uniform and the rapid change in metrics is unimportant because the solution isn't changing. However, if another variable such as velocity was being calculated in this region it would be extremely difficult to obtain an accurate solution for that variable. In this case the other variables (besides temperature) should be considered in the formation of the weight function and the grid transformation. A possible solution to this problem is shown in Fig. 9 where the grid distribution has been frozen at the value it had when the flame left the computational region. With this strategy the metrics are smooth but the mesh is wastefully fine near the outer boundary.

Another potential problem exists when different regions of high gradient exist within the same problem. This is particularly troublesome when the regions have incompatible

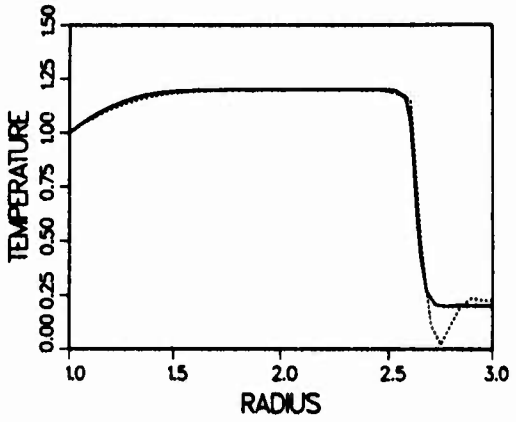

Figure 8. Typical temperature distribution with uniform and adaptive grids. Dashed line is on uniform grid, and shows oscillations due to high cell Peclet number.

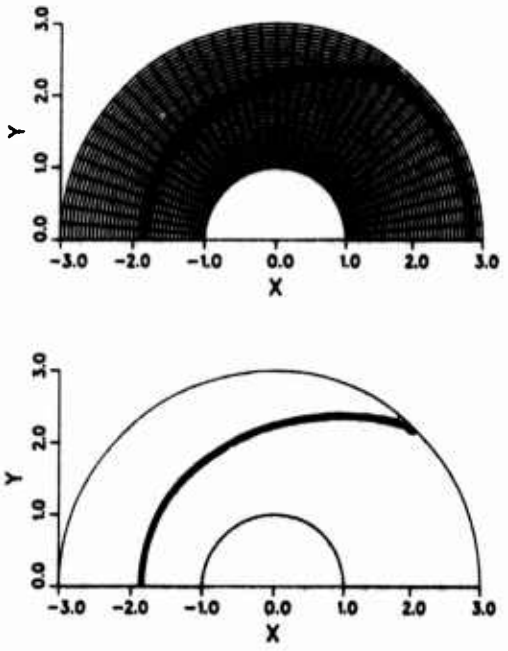

Figure 9. Coordinate system and isotherm distribution about a burning particle with a coordinate transformation adaptive grid. Coordinate system frozen at outer boundary.

geometries for grid stretching, as is the case in our next example. Here a flame surrounds a burning spherical particle over which the flow (Reynolds number of 100) has separated. In this calculation both the flow field in separation region and the temperature gradients in the flame must be resolved, and the boundary conditions must be applied far from the body. The coordinate system used for the flame is not well suited for the flow, and we have taken the approach of using two different coordinate systems and interpolating between them. Figure 10 shows the vorticity pattern together with the grid used to compute the flow field. The temperature distribution and its grid are shown in Fig. 11. Certainly the use of two coordinate systems increases storage and computation time, but the one order of magnitude improvement of grid resolution achieved by the adaptive gridding method, more than makes up for the additional effort. However, it is easily seen that this approach to grid adaptation introduces many new problems, which should prove fertile ground for new solution procedures.

CONCLUSIONS

We believe that we have achieved considerable success in applying adaptive grid methods to solve a variety of problems, but it is also true that the results are not complete. We have

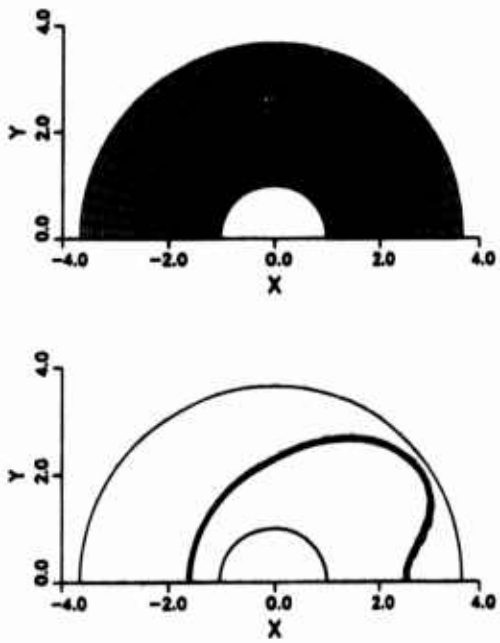

Figure 10. Coordinate system and isotherm distribution about a burning particle with separated flow, using a coordinate transformation adaptive grid.

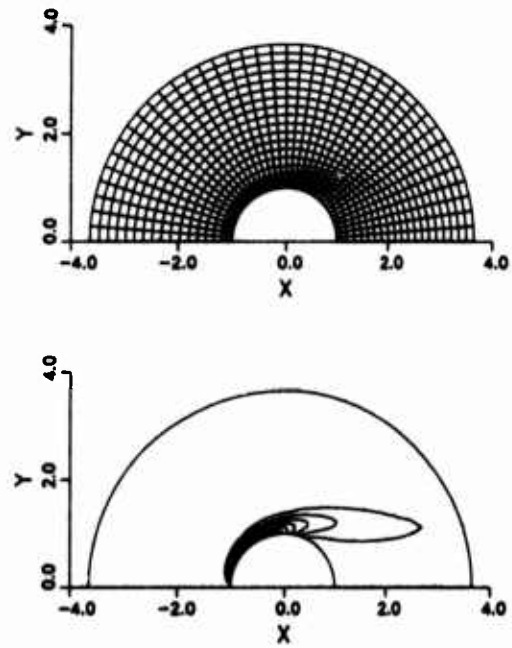

Figure 11. Coordinate system and vorticity distribution about a burning particle with separated flow, using a coordinate transformation adaptive grid.

seen clearly that adaptive gridding is necessary for a wide range of heat and mass transfer applications. Our examples demonstrated a strong dependence of flame shape, flame speed, and numerical stability on the mesh spacing. In regions such as flame fronts, the grid has to be so fine that uniform meshing is completely impractical. However, using the adaptive approach, we have kept cell Reynolds and Peclet numbers less than one with relatively few grid points!

We discussed the concept of equidistribution of a positive weight function and we regard it as a useful framework from which to develop adaptive grid methods. The variable node approach is analogous to ODE initial value problem software, and, because it attempts to bring the weight function within pre-specified bounds, it is potentially the most accurate approach to adaptive meshing. However, due to the number of grid points which may be needed, it is often inefficient in computer time and storage. On the other hand, using generalized coordinates and adaptive gridding through coordinate transformations allows for good resolution of body shapes and flame structure in separated flows at moderate Reynolds Number. In this case, however, the weight function is equidistributed, but not driven below a prespecified bound. We have successfully applied both the coordinate transformation approach and the variable node approach in one- and two-dimensions. Full two-dimensional generalizations are yet to come.

Although adaptive gridding is required for accurate resolution in many problems its use

can introduce new problems. For example, we see problems caused when high gradient regions intersect boundaries or leave the computational zone by convective processes. Also, when more than one physical variable causes scaling problems, such as in flame propagation and flow separation, it may be difficult to use one grid system for the entire problem. Instead, it may be avantageous to use more than one adaptive coordinate system simultaneously. So far the "fix" to many of these problems has been problem dependent. Generalizations are needed.

REFERENCES

1. Thompson, J. F., Thames, F. C. and Mastin C. M. (1974) *J. Comp. Phys.*, **15**, pp. 299-319.
2. Steger, J. L. (1978) *AIAA J.*, **16**, pp. 679-686.
3. Smooke, M. D. (1982) "Solution of Burner-Stabilized Pre-Mixed Laminar Flame Problems by Boundary Value Methods", Sandia National Laboratories Report, SAND81-8040.
4. Kautsky, J. and Nichols, N. K. (1979) "Equidistributing Meshes with Constraints," Stanford University Report, STAN-CS-79-766.
5. White, A. B. (1979) *SIAM J. Numer. Anal.*, **16**, pp. 472-502.
6. Pereyra, V. and Sewell, E. G. (1975) it Num. Math.,**23**, pp. 261-268.
7. Russell, R. D. and Christiansen, J. (1978) it SIAM J. Numer. Anal., **15**, pp. 59-80.
8. Ablow, C. M. and Schecter, S. (1978) *J. Comp. Phys.*,**27**, pp.351-362.
9. deRivas, E. K. (1972) *J. Comp. Phys*,**10**, pp. 202-210.
10. Denny, V. E. and Landis, R. B. (1972) *J. Comp. Phys*,**9**, pp. 120-137.
11. Pearson, C. E. (1968) *J. Math. Phys.*,**47**, pp. 134-154.
12. Smooke, M. D., (1982) "On the Use of a Modified Newton Method for the Solution of Boundary Value Problems", in preparation.
13. Berger, M., Gropp, W. D. and Oliger, J. (1981) "Grid Generation for Time Dependent Problems," NASA Conference Publication 2611.
14. Dwyer, H. A., Kee, R. J. and Sanders, B. R. (1980) *AIAA J.*,**18**, (10), pp 205-212.
15. Beam, R. M. and Warming, R. F. (1978) *AIAA J.*,**16**, pp. 393-402.
16. Briley, W. R. and McDonald, H. (1977) *J. Comp. Phys.*, **24** (4), pp. 372-397.
17. Kee, R. J. and Dwyer, H. A. (1982) *Progress in Astronautics and Aeronautics*,**76**, pp. 485-500, AIAA, New York.
18. Smooke, M. D., Miller, J. A. and Kee, R. J. (1982) "Numerical Solution of Burner-Stabilized Laminar Pre-Mixed Flame Problems, by an Efficient Boundary Value Method," to appear Proceedings of the GAMM-Conference on Pre-mixed Laminar Flames, Aachen, Germany.
19. Smooke, M. D., Miller, J. A. and Kee, R. J. (1982) "On the Use of Adaptive Grids in Numerically Calculating Adiabatic Flame Speeds," to appear, Proceedings of the GAMM-Conference on Pre-mixed Laminar Flames, Aachen, Germany.
20. Roache, P. (1971) "Computational Fluid Dynamics," Hermosa Publishers, Albuquerque, NM.